zmPDSwR Chapter 3 Part I

coop711

2015년 9월 12일

Data

자료 읽어들이기

```
setwd("~/Dropbox/Works/Class/Data_Science/R.WD/zmPDSwR/")
custdata <- read.table("../../zmPDSwR/Custdata/custdata.tsv", header=TRUE, se
p="\t", stringsAsFactors=TRUE)
```

기초 통계

- Missing values는 어디에 많이 등장하는가? 그 이유는 무엇이라고 생각되는가?

```
summary(custdata)
```

```
##      custid        sex     is.employed         income
##  Min.   :   2068  F:440   Mode :logical   Min.   : -8700
##  1st Qu.: 345667  M:560   FALSE:73        1st Qu.: 14600
##  Median : 693403          TRUE :599       Median : 35000
##  Mean   : 698500          NA's :328       Mean   : 53505
##  3rd Qu.:1044606                          3rd Qu.: 67000
##  Max.   :1414286                          Max.   :615000
##
##              marital.stat health.ins
##  Divorced/Separated:155   Mode :logical
##  Married           :516   FALSE:159
##  Never Married     :233   TRUE :841
##  Widowed           : 96   NA's :0
##
##
##
##                        housing.type recent.move      num.vehicles
##  Homeowner free and clear    :157   Mode :logical   Min.   :0.000
##  Homeowner with mortgage/loan:412   FALSE:820       1st Qu.:1.000
##  Occupied with no rent       : 11   TRUE :124       Median :2.000
##  Rented                      :364   NA's :56        Mean   :1.916
##  NA's                        : 56                   3rd Qu.:2.000
##                                                     Max.   :6.000
##                                                     NA's   :56
##       age              state.of.res
##  Min.   :  0.0   California  :100
##  1st Qu.: 38.0   New York    : 71
##  Median : 50.0   Pennsylvania: 70
##  Mean   : 51.7   Texas       : 56
##  3rd Qu.: 64.0   Michigan    : 52
##  Max.   :146.7   Ohio        : 51
##                  (Other)     :600
```

- 타당치 않은 값들을 찾아본다면?

```
summary(custdata$income)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -8700   14600   35000   53500   67000  615000
```

```
summary(custdata$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    38.0    50.0    51.7    64.0   146.7
```

자료 구조

- `factor` 의 `class` , `mode` , `typeof` 가 각각 어떻게 나타나는지 유의

```
str(custdata)
```

```
## 'data.frame':    1000 obs. of  11 variables:
##  $ custid      : int  2068 2073 2848 5641 6369 8322 8521 12195 14989 15917
...
##  $ sex         : Factor w/ 2 levels "F","M": 1 1 2 2 1 1 2 2 2 1 ...
##  $ is.employed : logi  NA NA TRUE TRUE TRUE TRUE ...
##  $ income      : int  11300 0 4500 20000 12000 180000 120000 40000 9400 2400
0 ...
##  $ marital.stat: Factor w/ 4 levels "Divorced/Separated",..: 2 2 3 3 3 3 3 2
2 1 ...
##  $ health.ins  : logi  TRUE TRUE FALSE FALSE TRUE TRUE ...
##  $ housing.type: Factor w/ 4 levels "Homeowner free and clear",..: 1 4 4 3 4
2 1 4 4 1 ...
##  $ recent.move : logi  FALSE TRUE TRUE FALSE TRUE FALSE ...
##  $ num.vehicles: int  2 3 3 0 1 1 1 3 2 1 ...
##  $ age         : num  49 40 22 22 31 40 39 48 44 70 ...
##  $ state.of.res: Factor w/ 50 levels "Alabama","Alaska",..: 22 9 10 31 9 32
12 22 13 33 ...
```

```
sapply(custdata, class)
```

```
##       custid          sex  is.employed       income marital.stat
##    "integer"     "factor"    "logical"    "integer"     "factor"
##   health.ins housing.type  recent.move num.vehicles          age
##    "logical"     "factor"    "logical"    "integer"    "numeric"
## state.of.res
##     "factor"
```

```
sapply(custdata, mode)
```

```
##       custid          sex  is.employed       income marital.stat
##    "numeric"    "numeric"    "logical"    "numeric"    "numeric"
##   health.ins housing.type  recent.move num.vehicles          age
##    "logical"    "numeric"    "logical"    "numeric"    "numeric"
## state.of.res
##    "numeric"
```

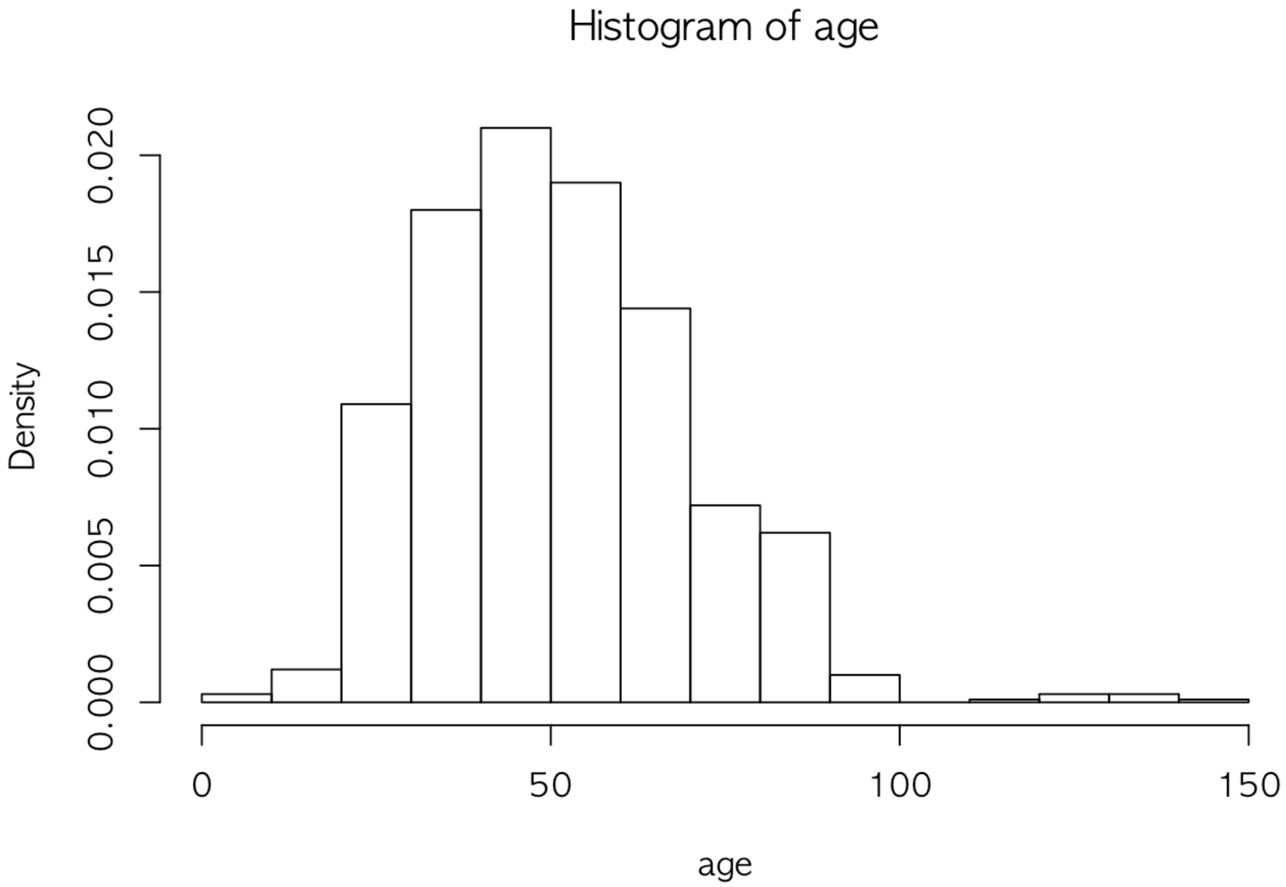
```
sapply(custdata, typeof)
```

```
##       custid          sex  is.employed       income marital.stat
##    "integer"    "integer"    "logical"    "integer"    "integer"
##   health.ins housing.type  recent.move num.vehicles          age
##    "logical"    "integer"    "logical"    "integer"     "double"
## state.of.res
##    "integer"
```

Visualization

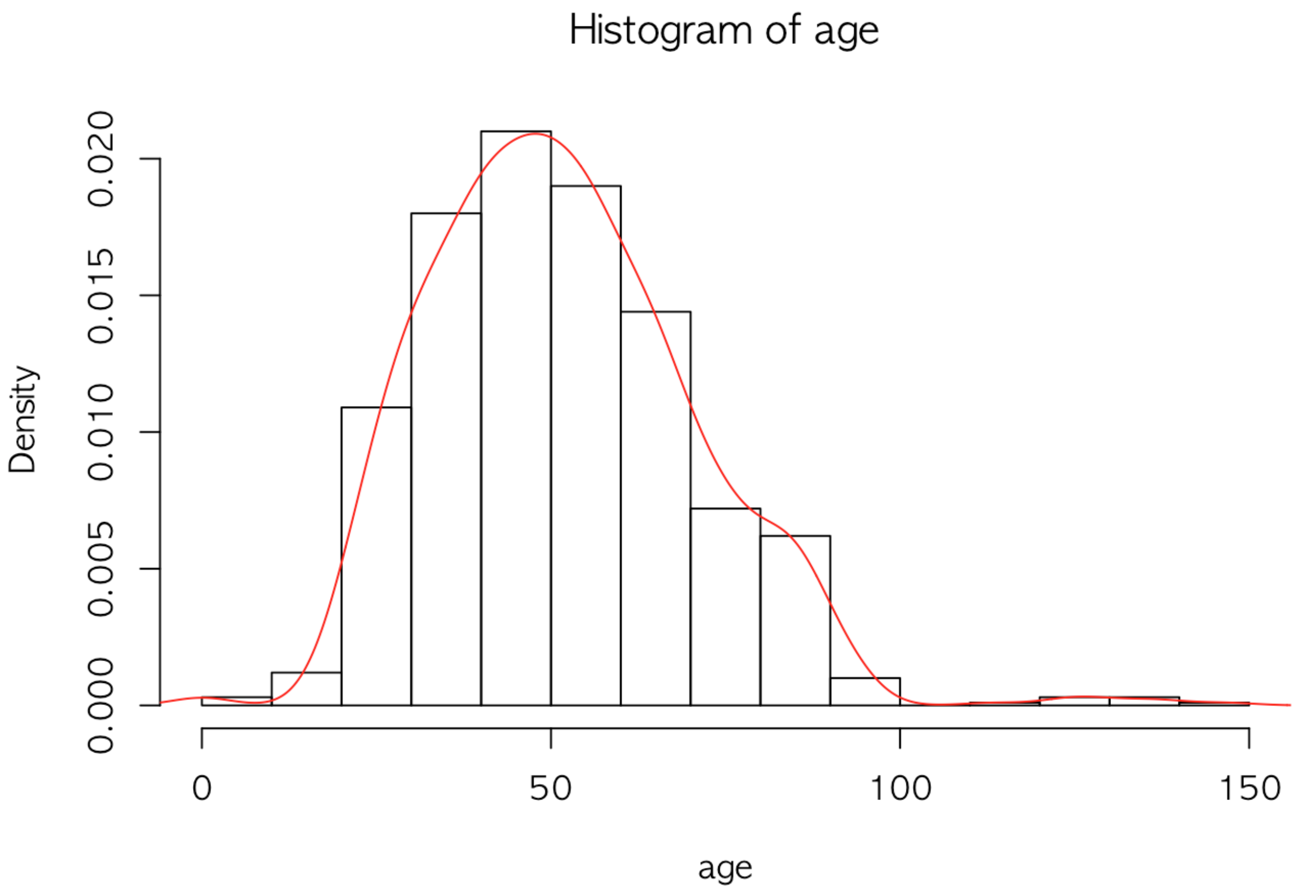
- `with` 를 쓰지 않고 `hist(custdata$age, prob=TRUE)` 로 하면 어느 요소가 어떻게 달라지는가?

```
with(custdata, hist(age, prob=TRUE))
```



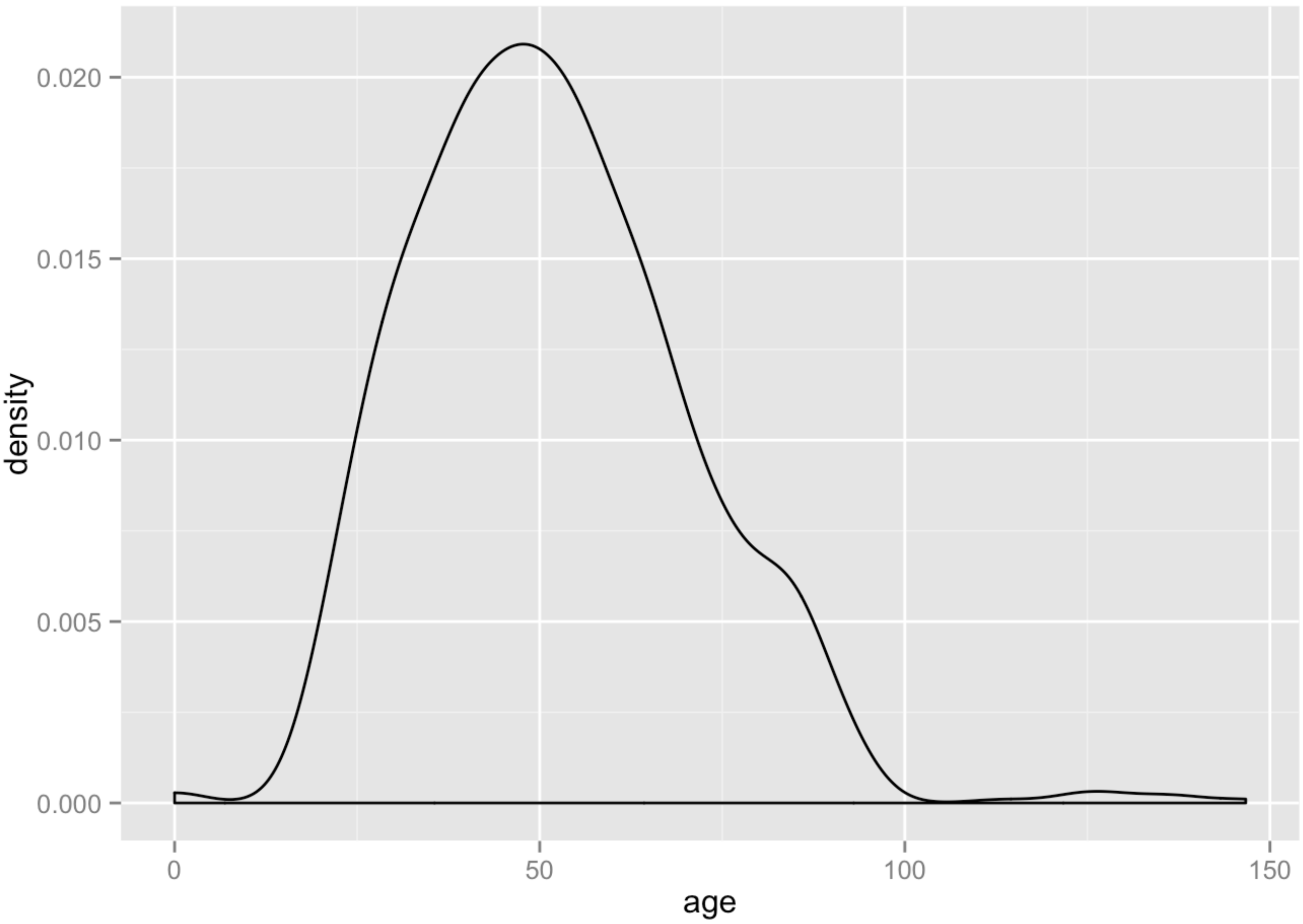
- `density estimation` 을 추가

```
with(custdata, hist(age, prob=TRUE))
with(custdata, lines(density(age), col="red"))
```



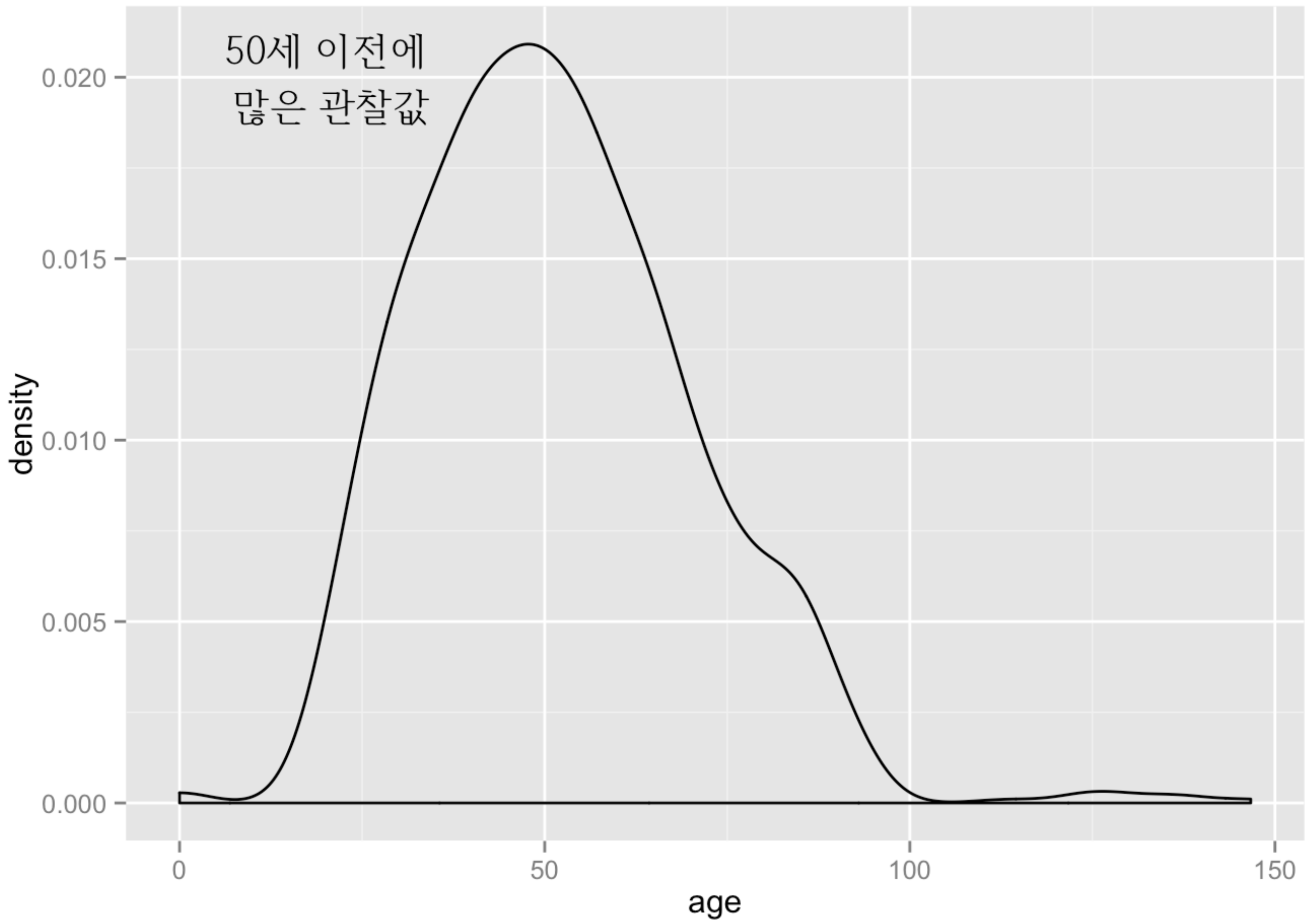
- ggplot 으로 표현하면,

```
library(ggplot2)
(g1 <- ggplot(custdata, aes(x=age)) + geom_density())
```

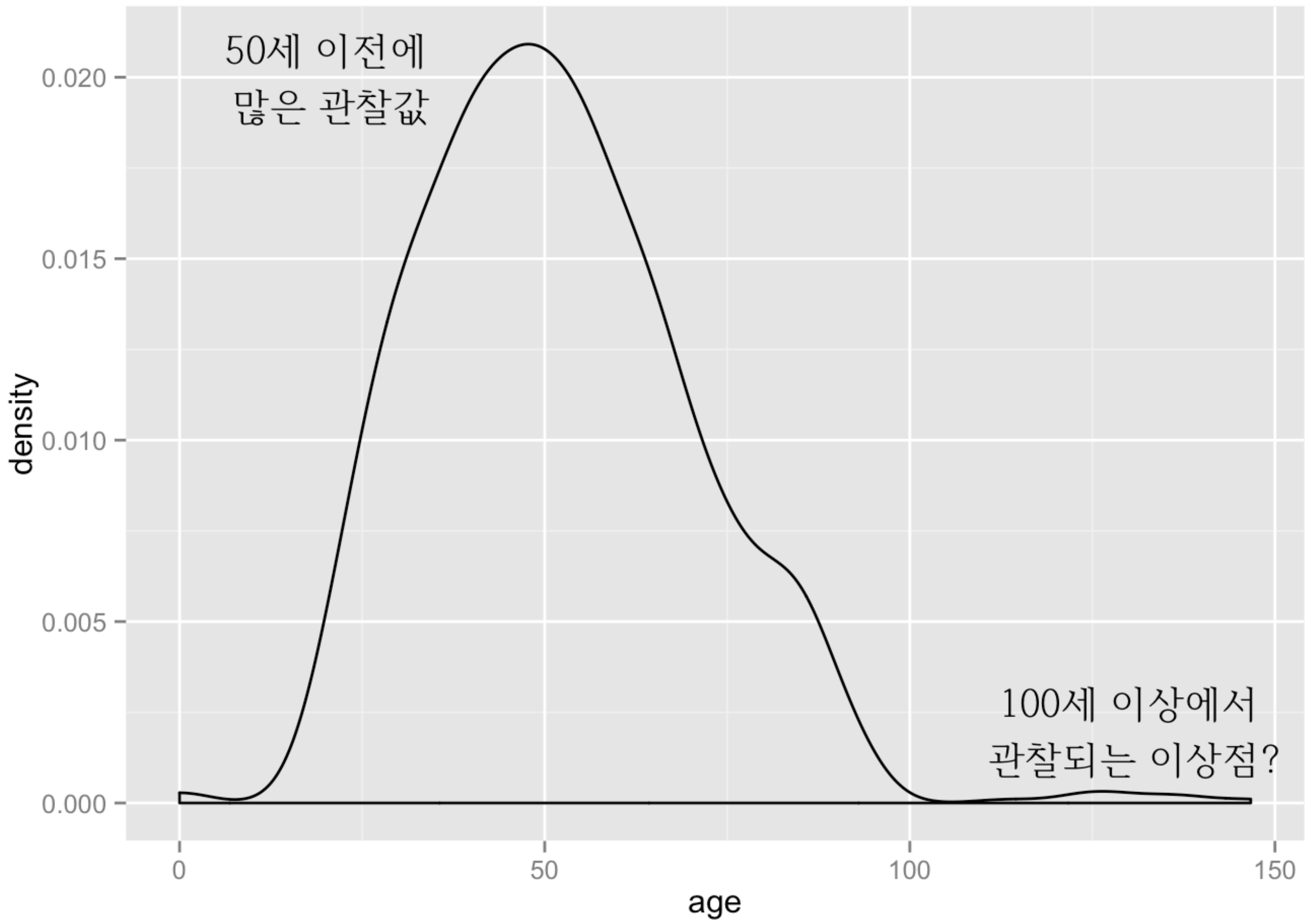


- 도표 안에 텍스트를 추가하려면, `annotate()` 사용

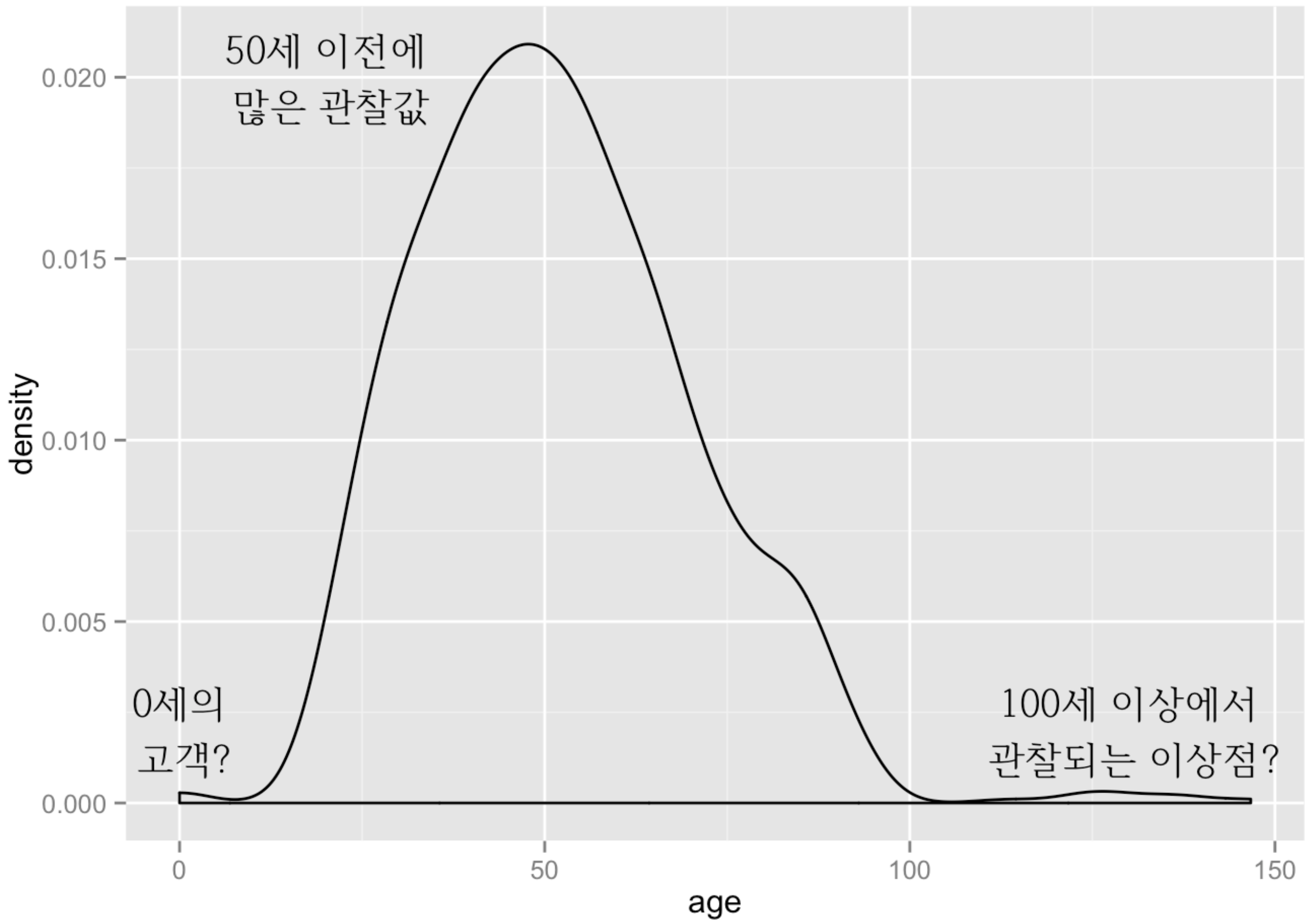
```
(g2 <- g1 + annotate("text", x=20, y=0.02, label="50세 이전에\n 많은 관찰값", famil
y="HCR Batang LVT"))
```



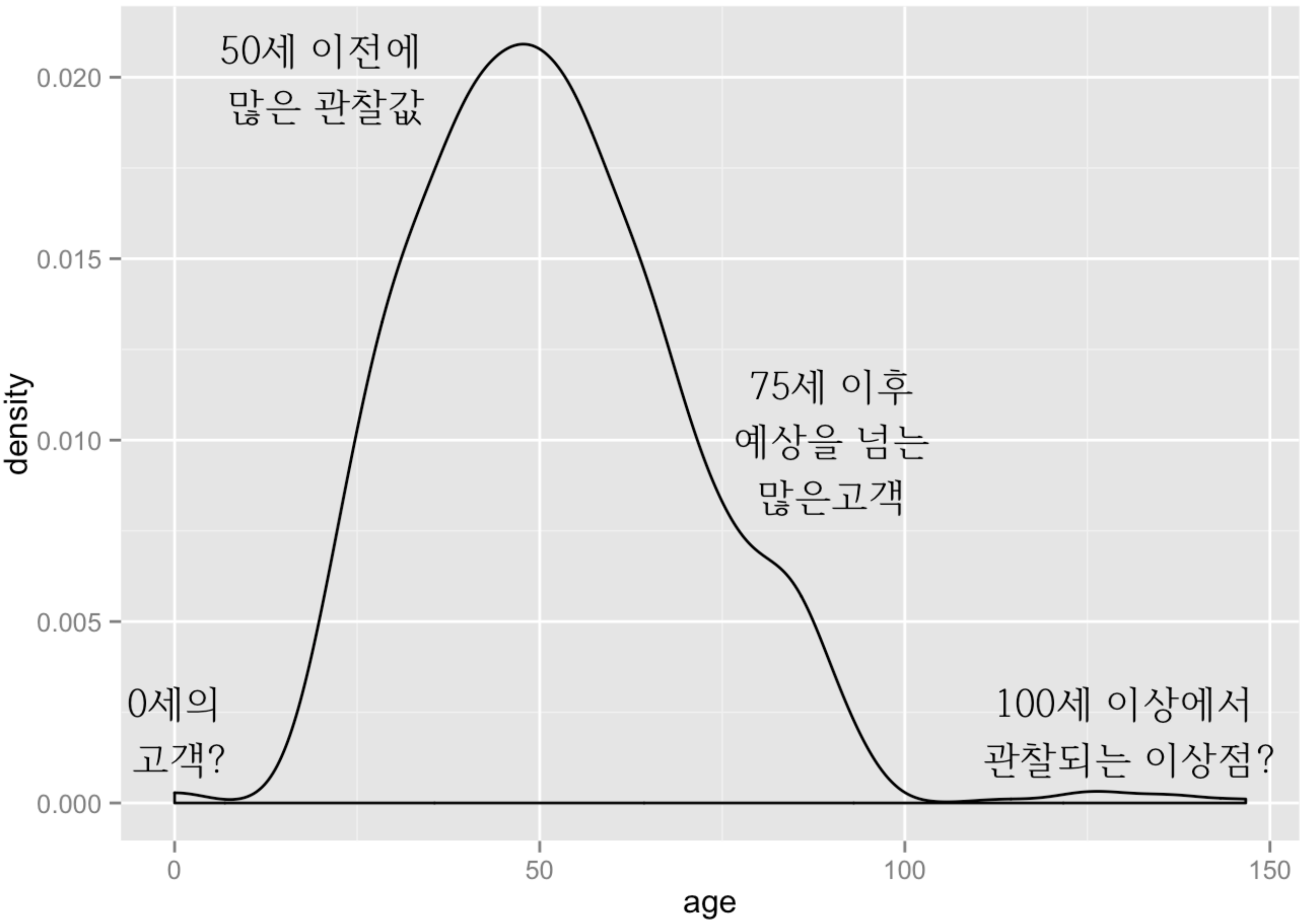
```
(g3 <- g2 + annotate("text", x=130, y=0.002, label="100세 이상에서\n 관찰되는 이상
점?", family="HCR Batang LVT"))
```



```
(g4 <- g3 + annotate("text", x=0, y=0.002, label="0세의\n 고객?", family="HCR Bat
ang LVT"))
```

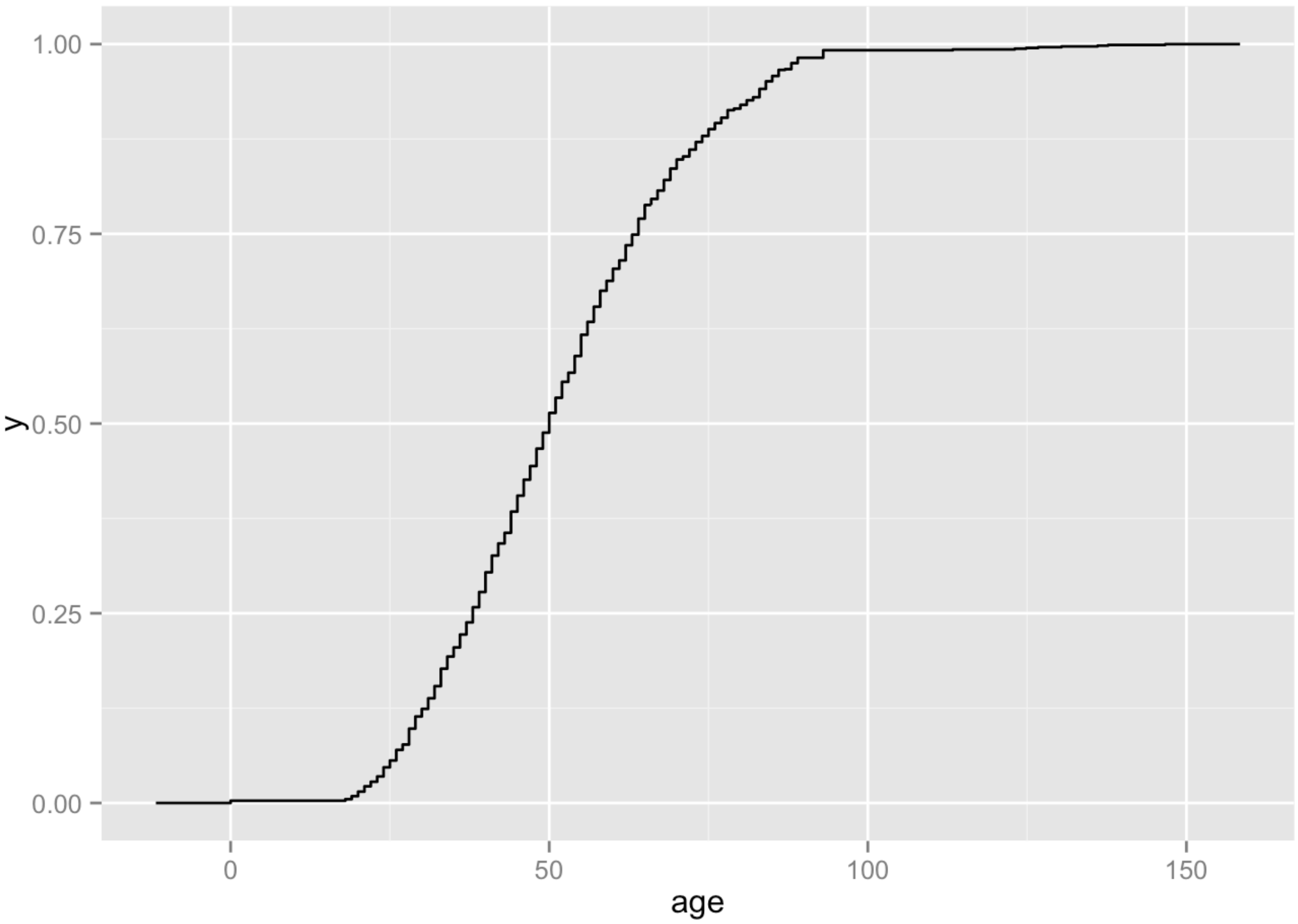


```
(g5 <- g4 + annotate("text", x=90, y=0.01, label="75세 이후\n예상을 넘는\n많은고객", f
amily="HCR Batang LVT"))
```



- 기초통계를 파악하는 데는 `summary()` 가 낫다는 기술에 대해서. 적어도 분위수에 관한 한 `ecdf` 가 시각적으로 우수함.

```
(g.ecdf <- ggplot(custdata, aes(x=age)) + stat_ecdf())
```

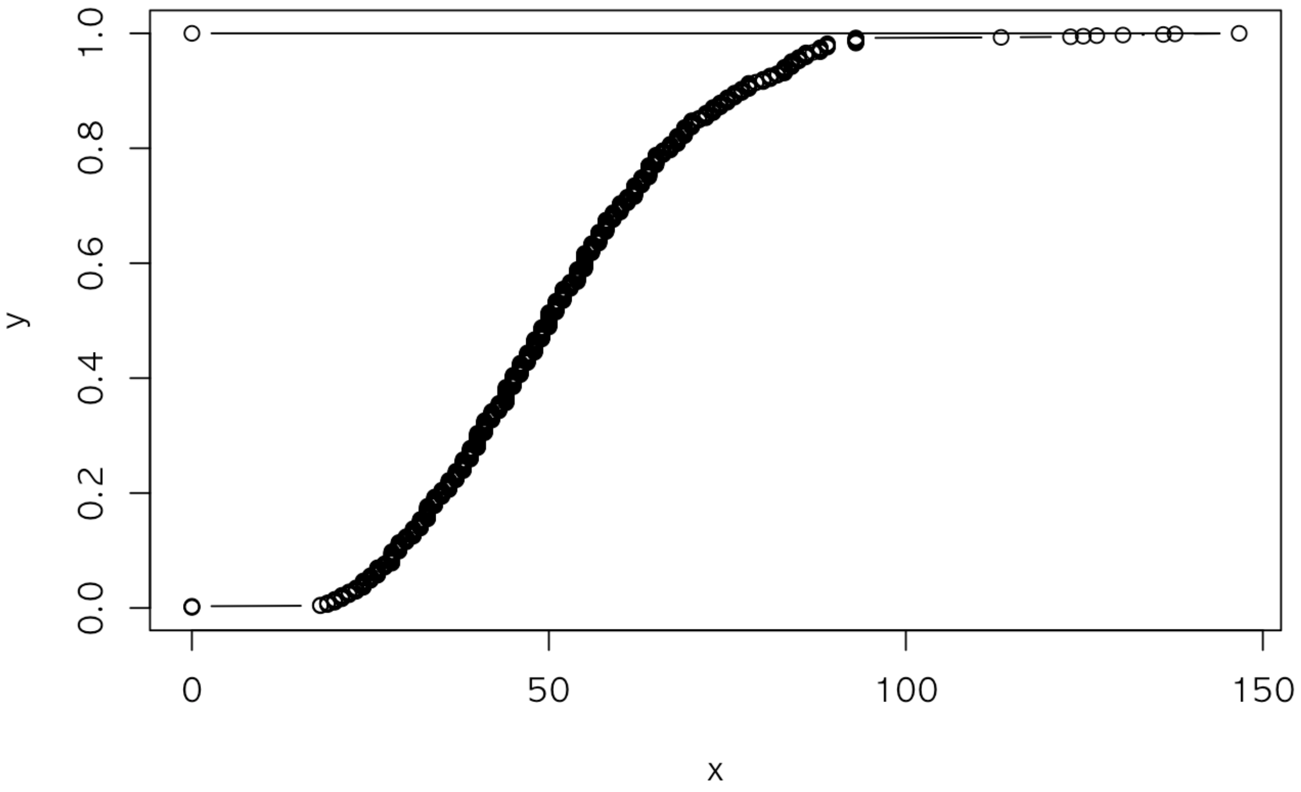


- 게다가 도표 윗 부분은 바로 평균이라는 점을 기억해 두어야 할 것임. 좀 복잡해 보이지만, `geom_polygon()` 을 이용하기 위해서는 다각형을 나타내는 좌표를 data frame으로 갖춰야 함.

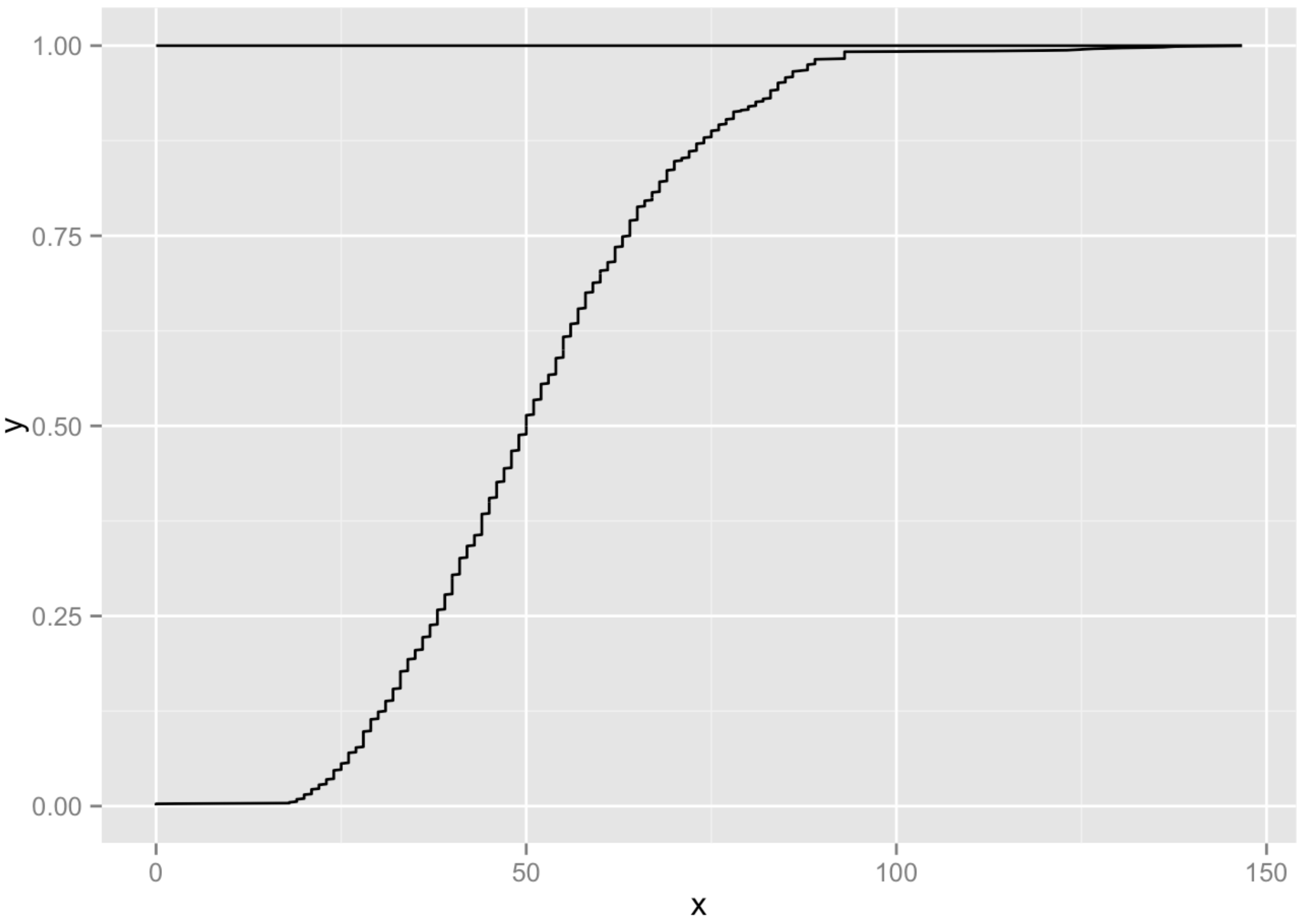
```
poly.x <- c(sort(custdata$age), sort(custdata$age)[1])
poly.y <- c((1:length(custdata$age))/length(custdata$age), 1)
poly.age <- data.frame(x=poly.x, y=poly.y)
```

- 제대로 갖추었는지 확인

```
plot(y ~ x, data=poly.age, type="b")
```

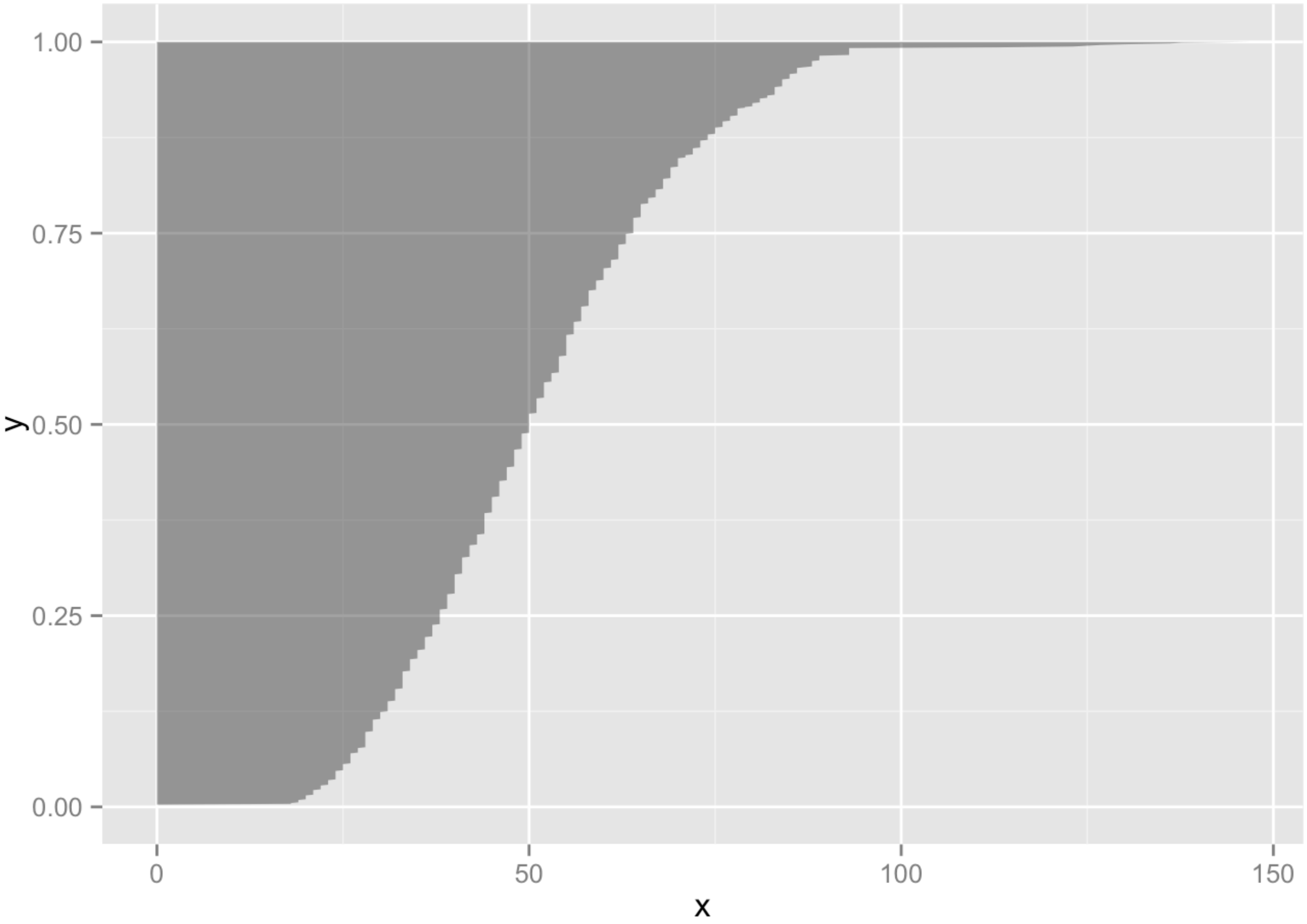


```
ggplot(poly.age, aes(x=x, y=y)) + geom_path()
```

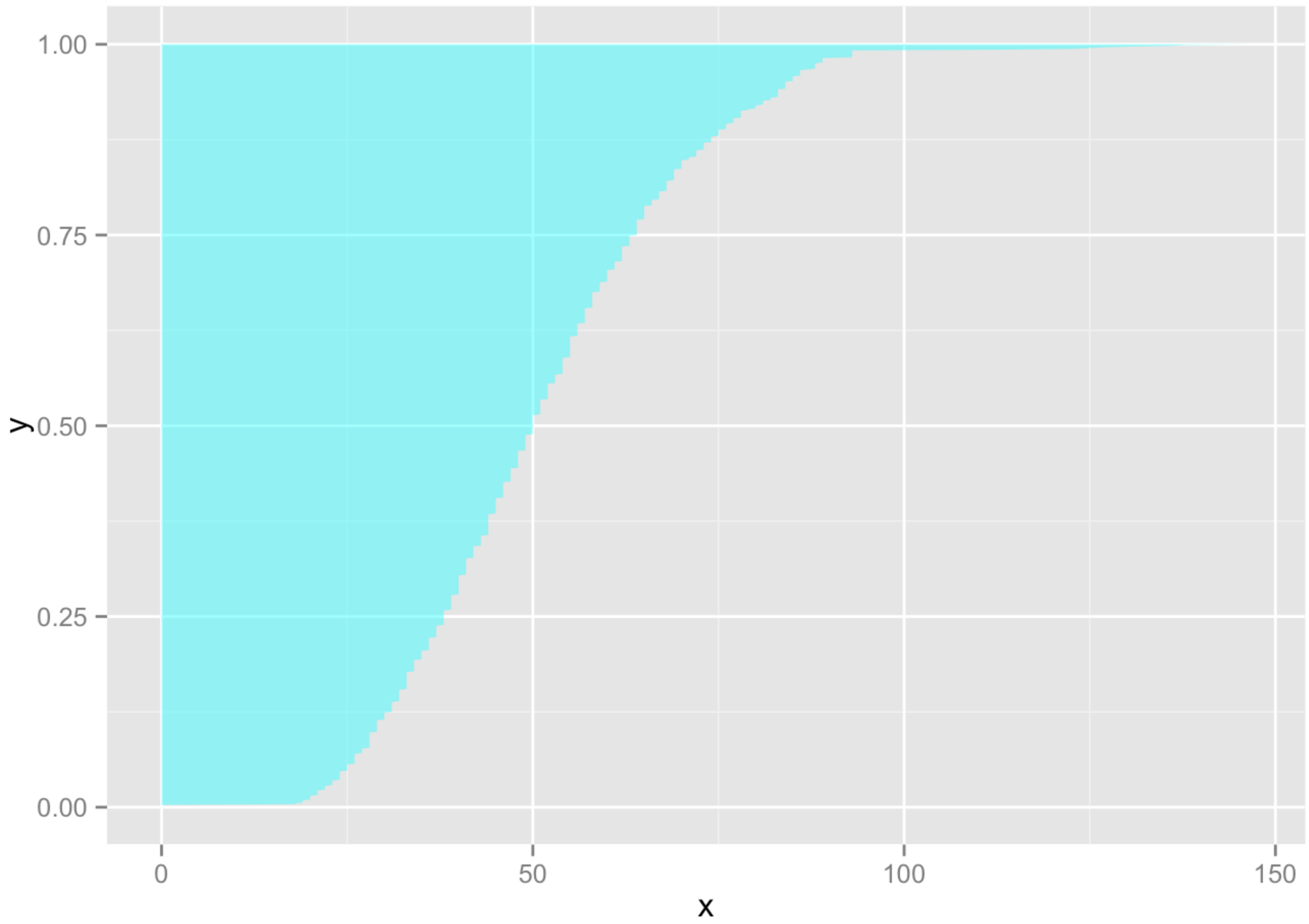


- `geom_polygon()` 에 `alpha` 로 조정. 색은 `fill` 로 설정.

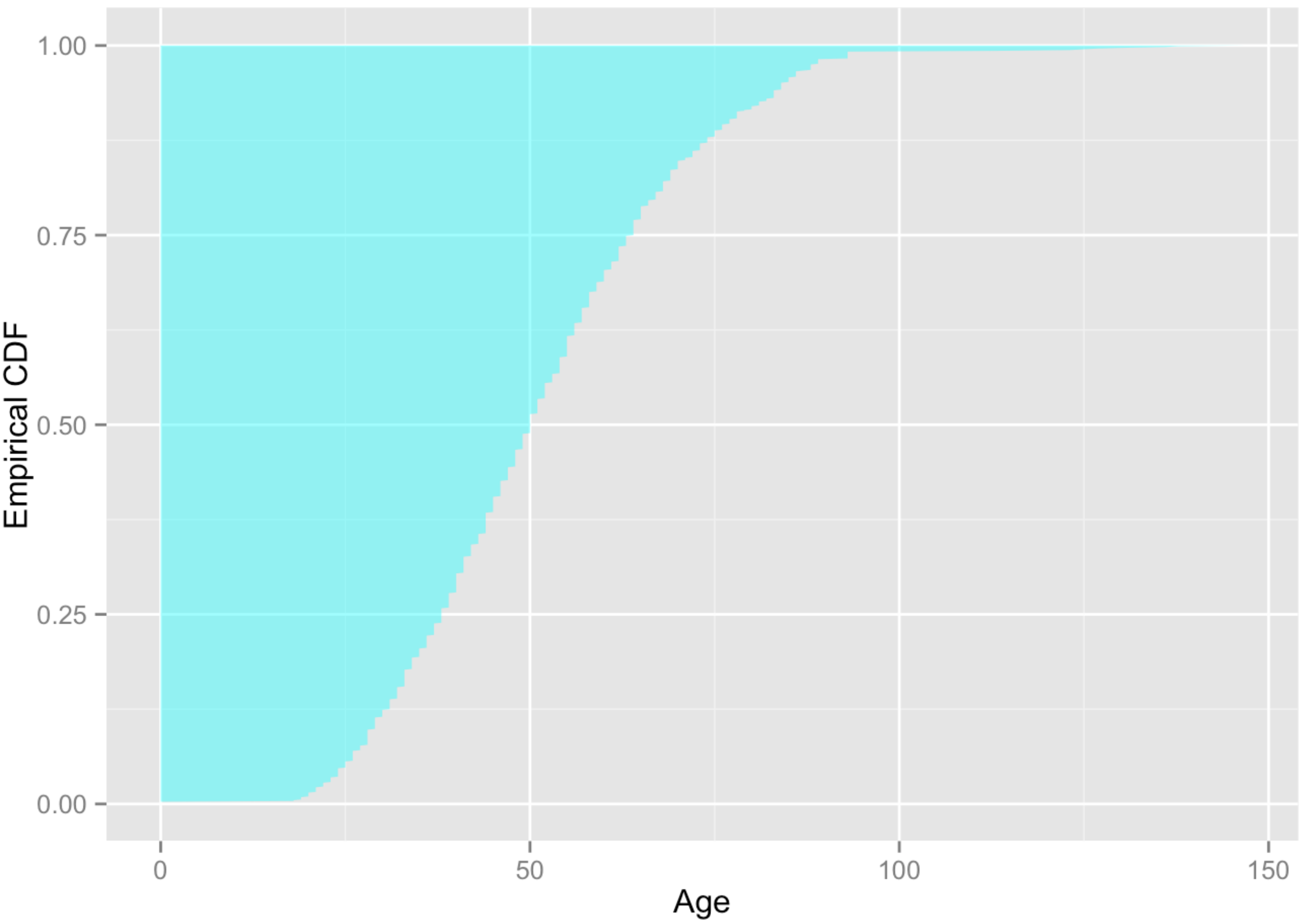
```
(p <- ggplot(poly.age, aes(x=x, y=y)) + geom_polygon(alpha=0.5))
```



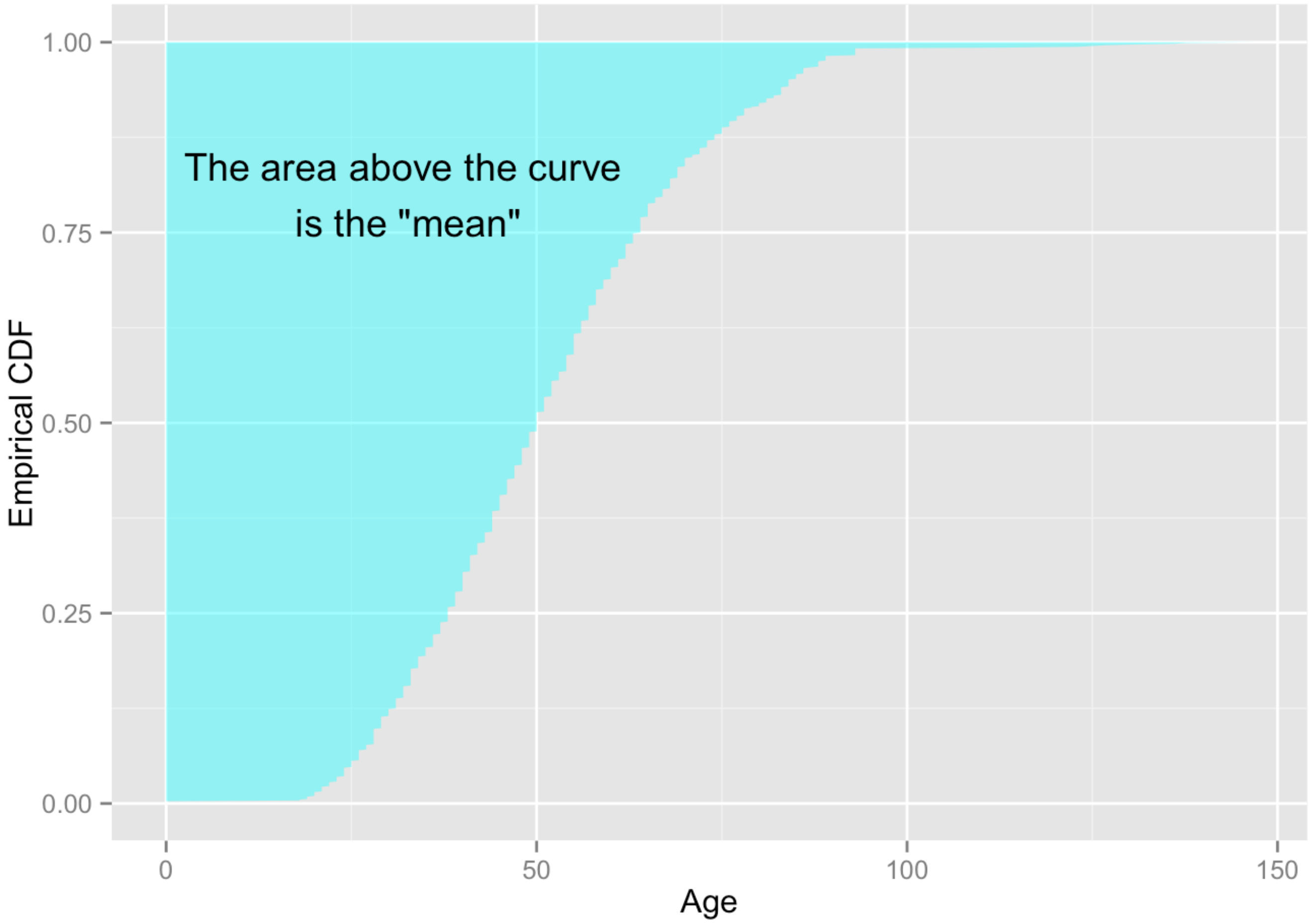
```
(p1<- ggplot(poly.age, aes(x=x, y=y)) + geom_polygon(fill="cyan", alpha=0.5))
```



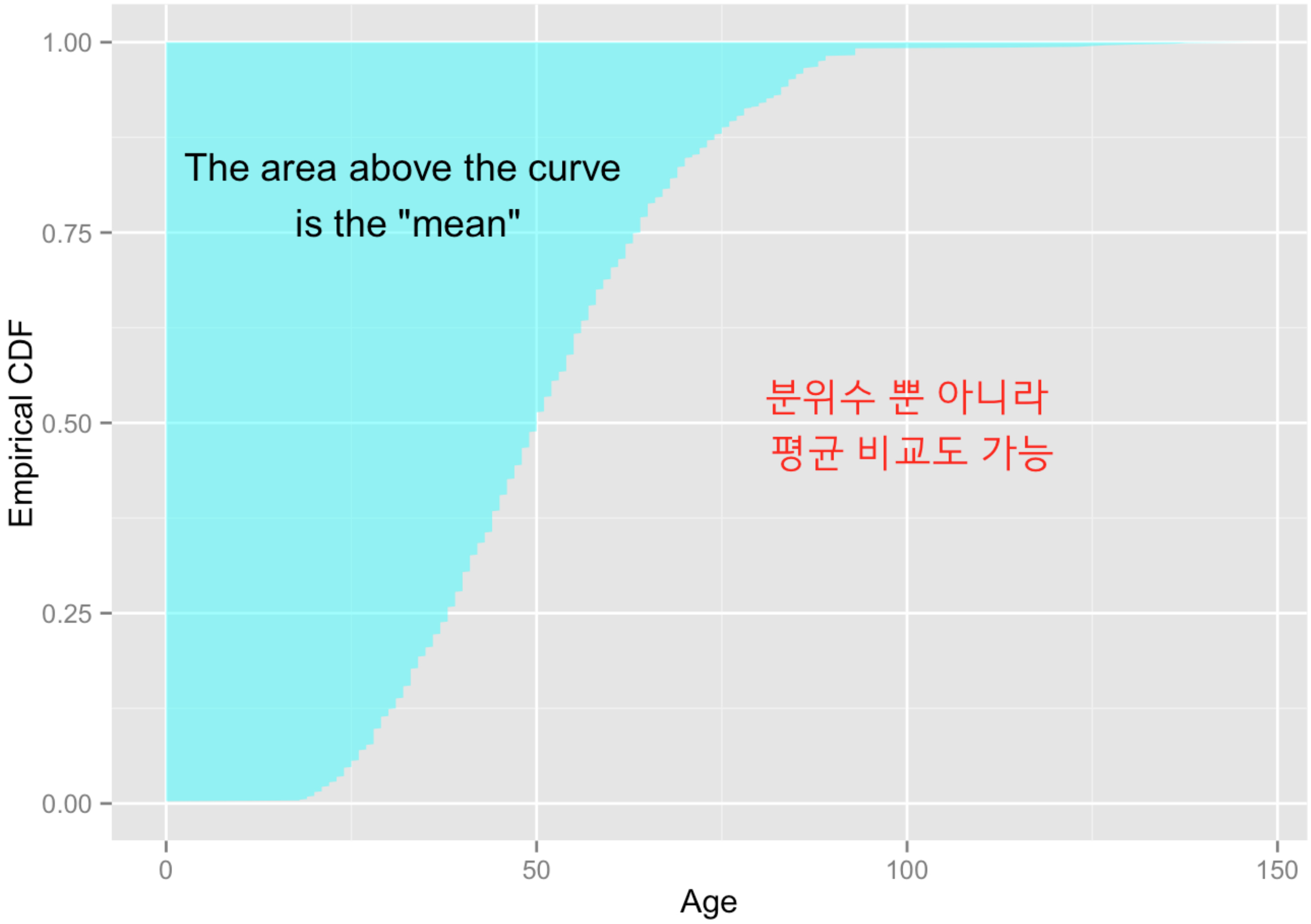
```
(p2 <- p1 + xlab("Age") + ylab("Empirical CDF"))
```



```
(p3 <- p2 + annotate("text", x=32, y=0.8, label="The area above the curve\n is
the \"mean\""))
```

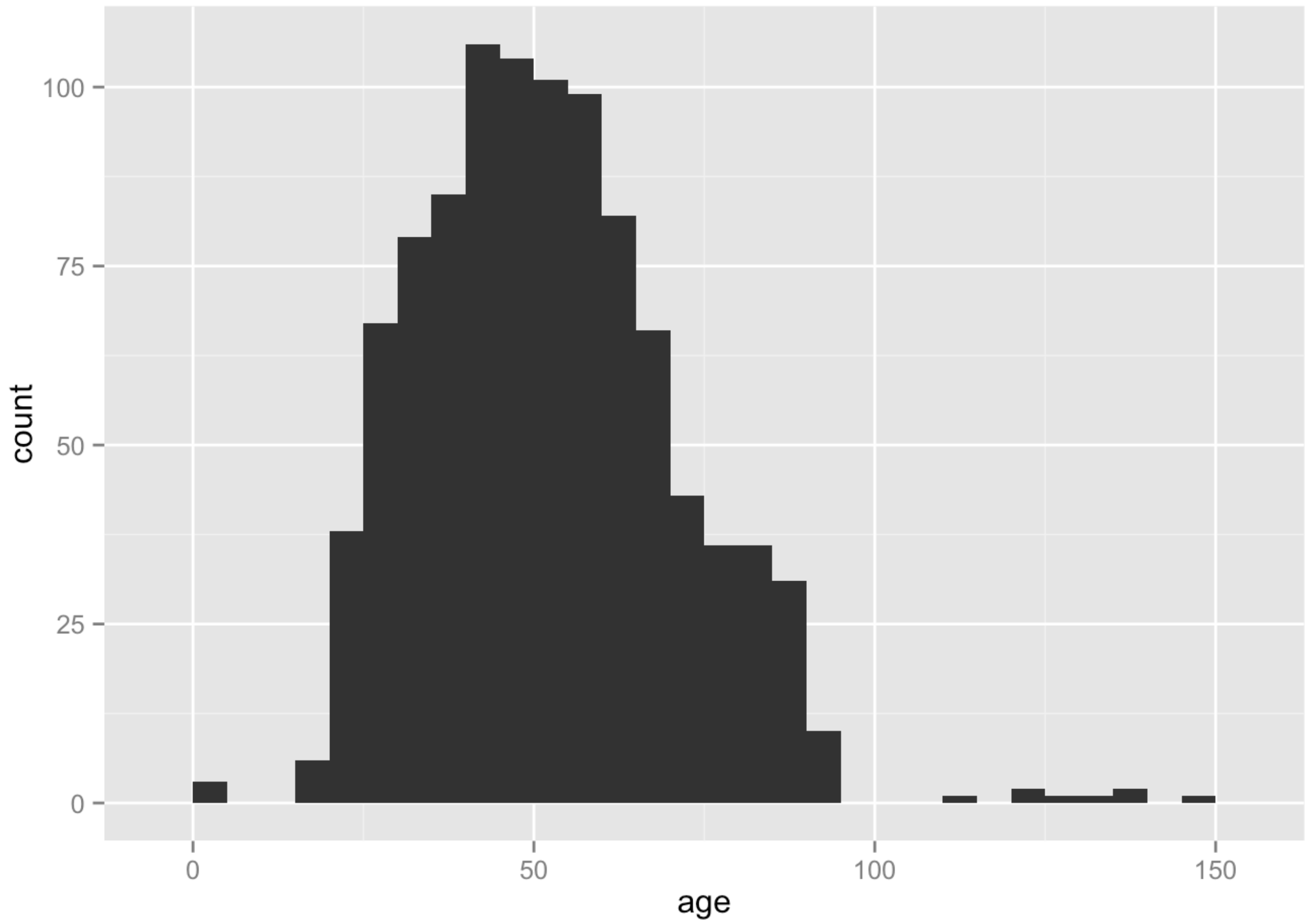


```
(p4 <- p3 + annotate("text", x=100, y=0.5, label="분위수 뿐 아니라\n 평균 비교도 가능",
family="HCR Dotum LVT", colour="red"))
```

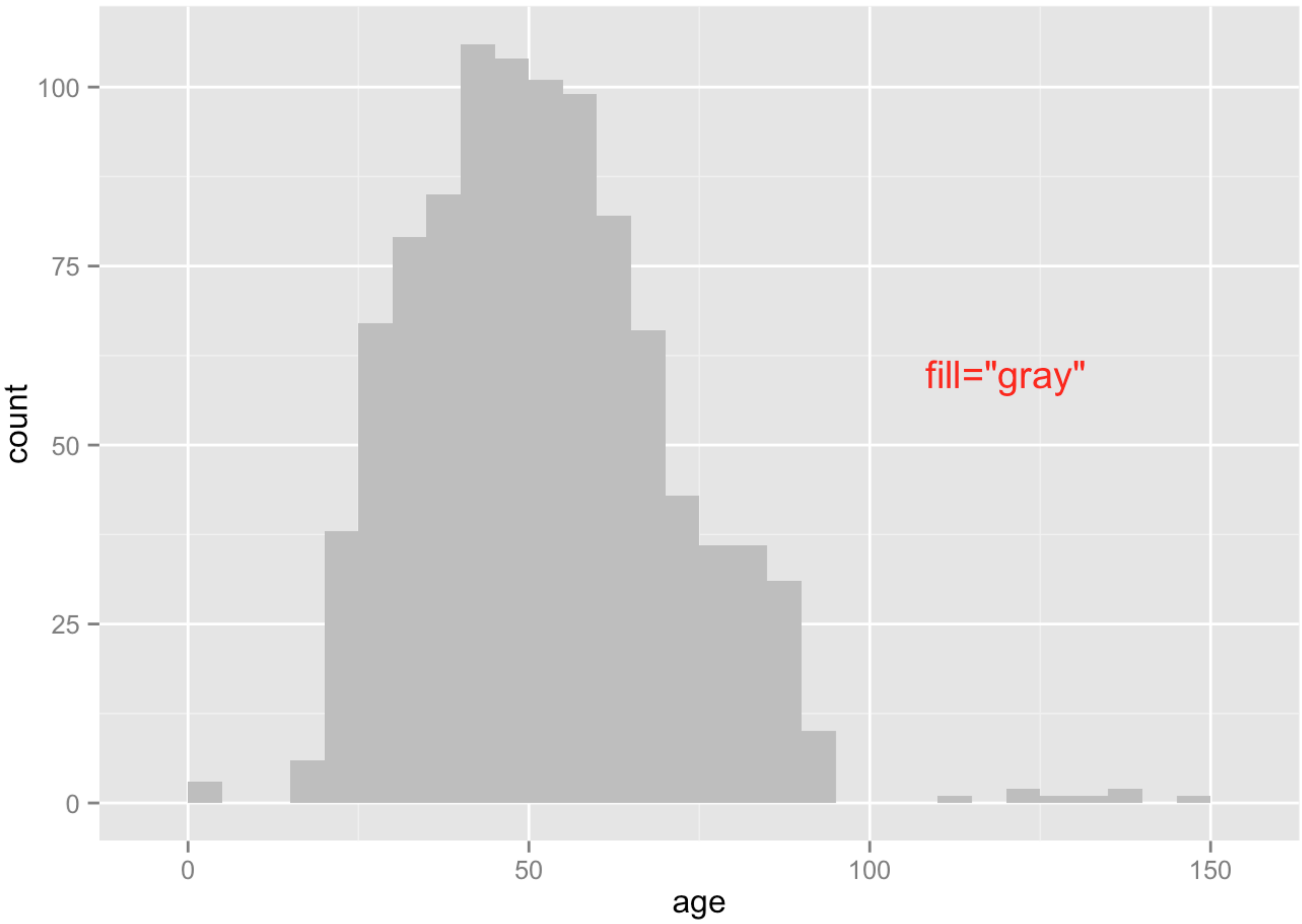


- 히스토그램으로 요약하기. 각각의 차이가 어디서 비롯되는지 이해할 것.

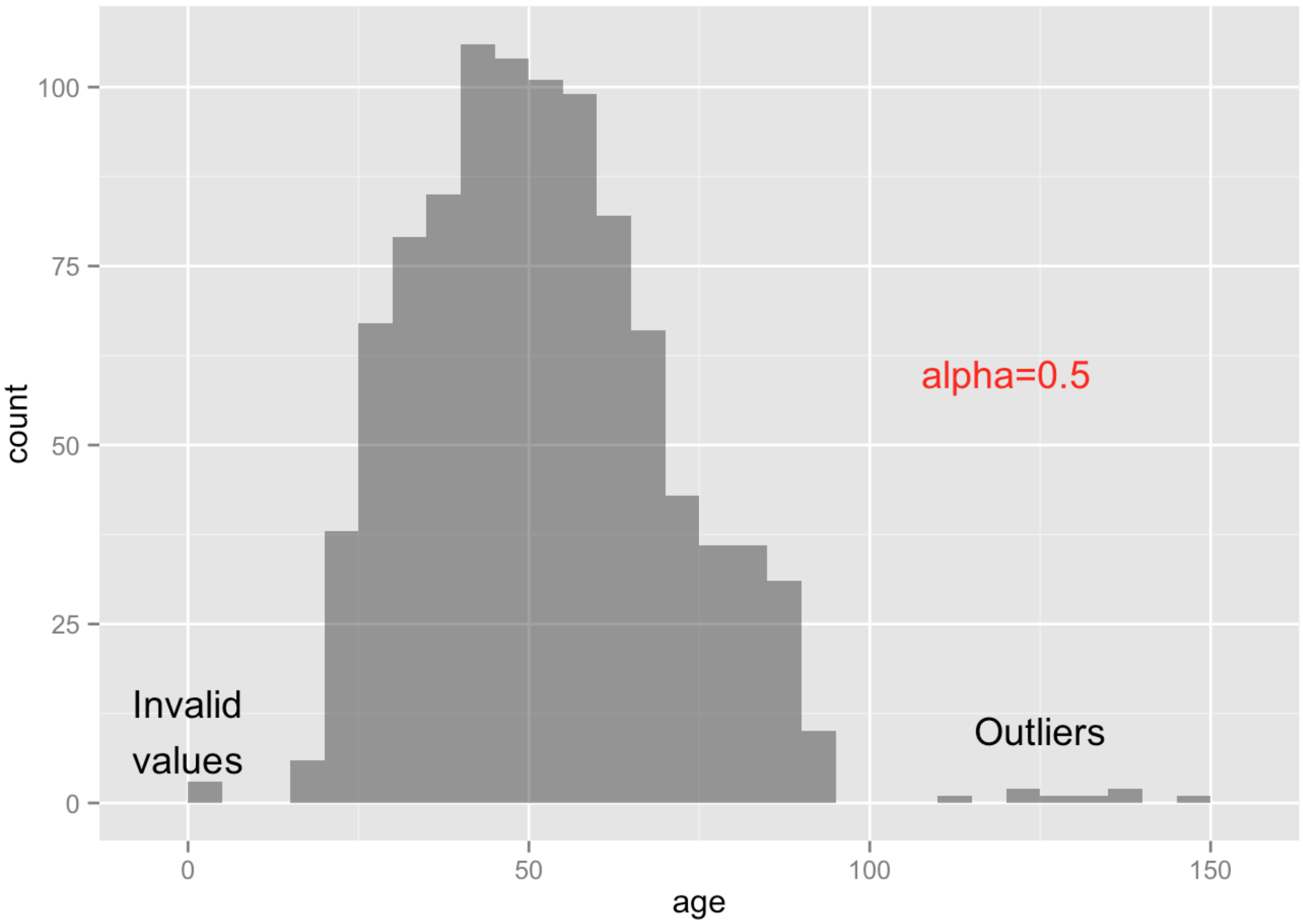
```
ggplot(custdata, aes(x=age)) + geom_histogram(binwidth=5)
```



```
ggplot(custdata, aes(x=age)) + geom_histogram(binwidth=5, fill="gray") +
  annotate("text", x=120, y=60, label="fill=\"gray\"", colour="red")
```

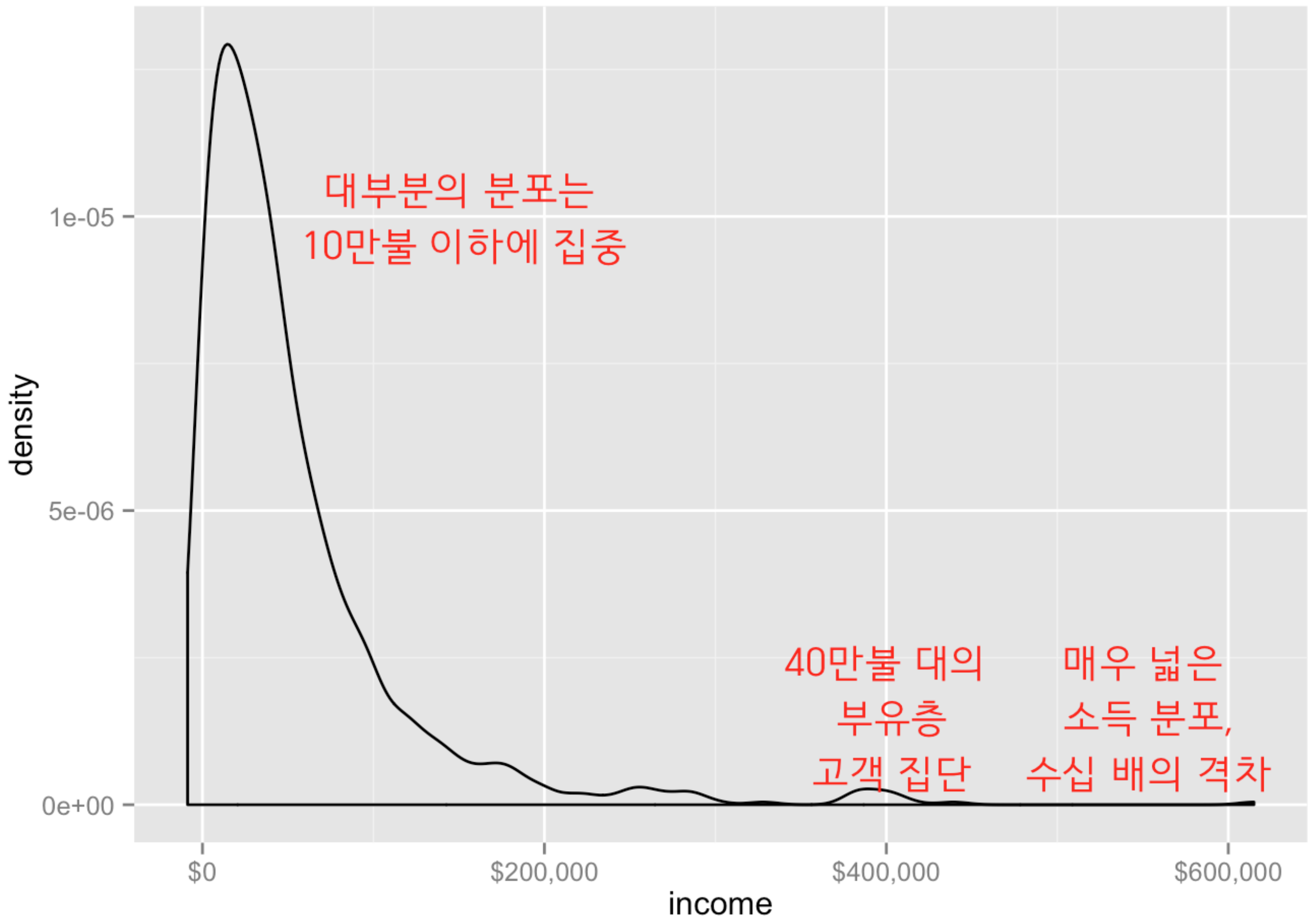


```
ggplot(custdata, aes(x=age)) + geom_histogram(binwidth=5, alpha=0.5) +
  annotate("text", x=120, y=60, label="alpha=0.5", colour="red") +
  annotate("text", x=125, y=10, label="Outliers") +
  annotate("text", x=0, y=10, label="Invalid\nvalues")
```



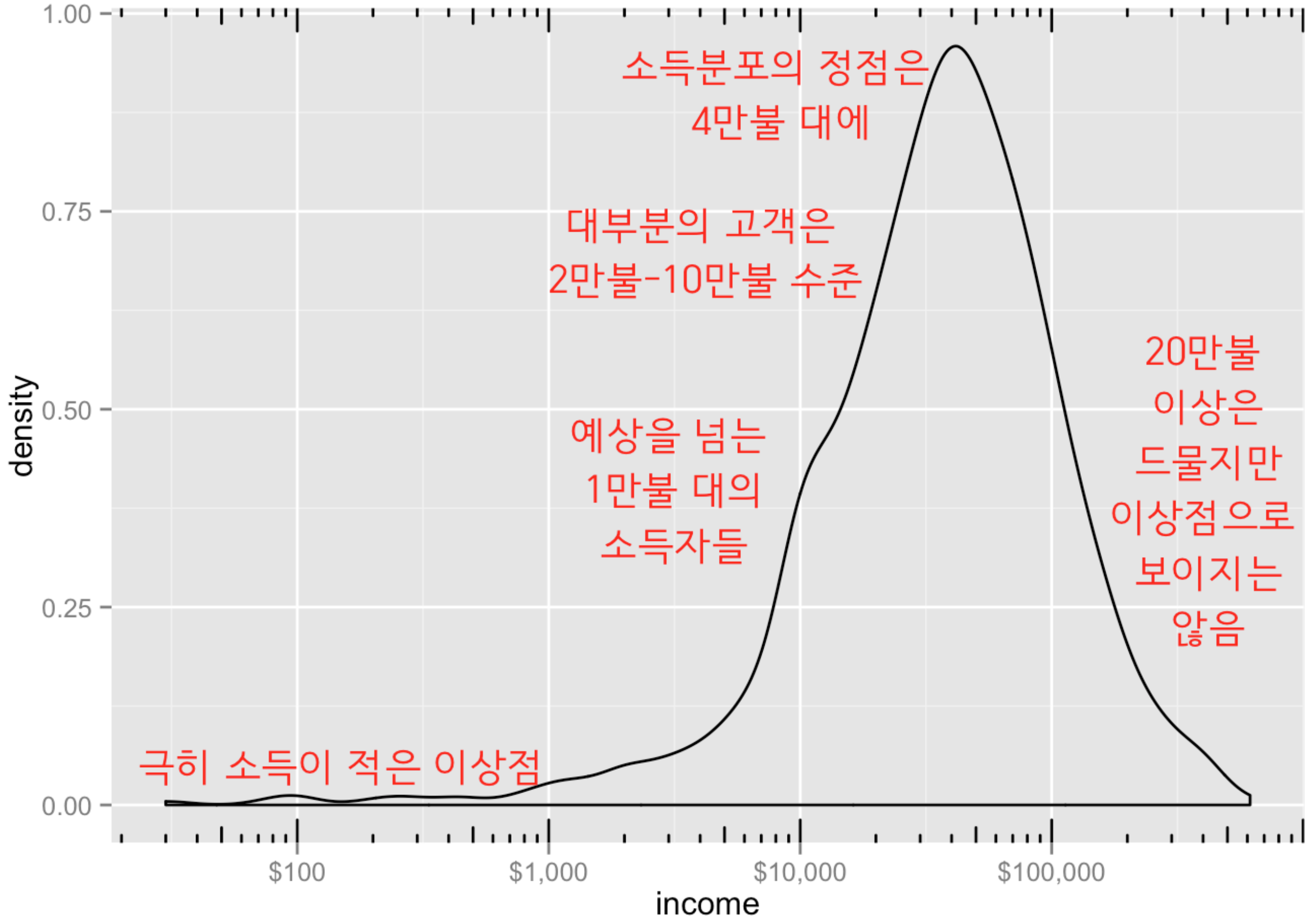
- Density Plots

```
library(scales)
ggplot(custdata) + geom_density(aes(x=income)) +
  scale_x_continuous(labels=dollar) +
  annotate("text", x=150000, y=0.00001, label="대부분의 분포는\n 10만불 이하에 집중", f
amily="HCR Dotum LVT", colour="red") +
  annotate("text", x=400000, y=0.0000015, label="40만불 대의\n 부유층\n 고객 집단", f
amily="HCR Dotum LVT", colour="red") +
  annotate("text", x=550000, y=0.0000015, label="매우 넓은\n 소득 분포,\n 수십 배의 격
차", family="HCR Dotum LVT", colour="red")
```



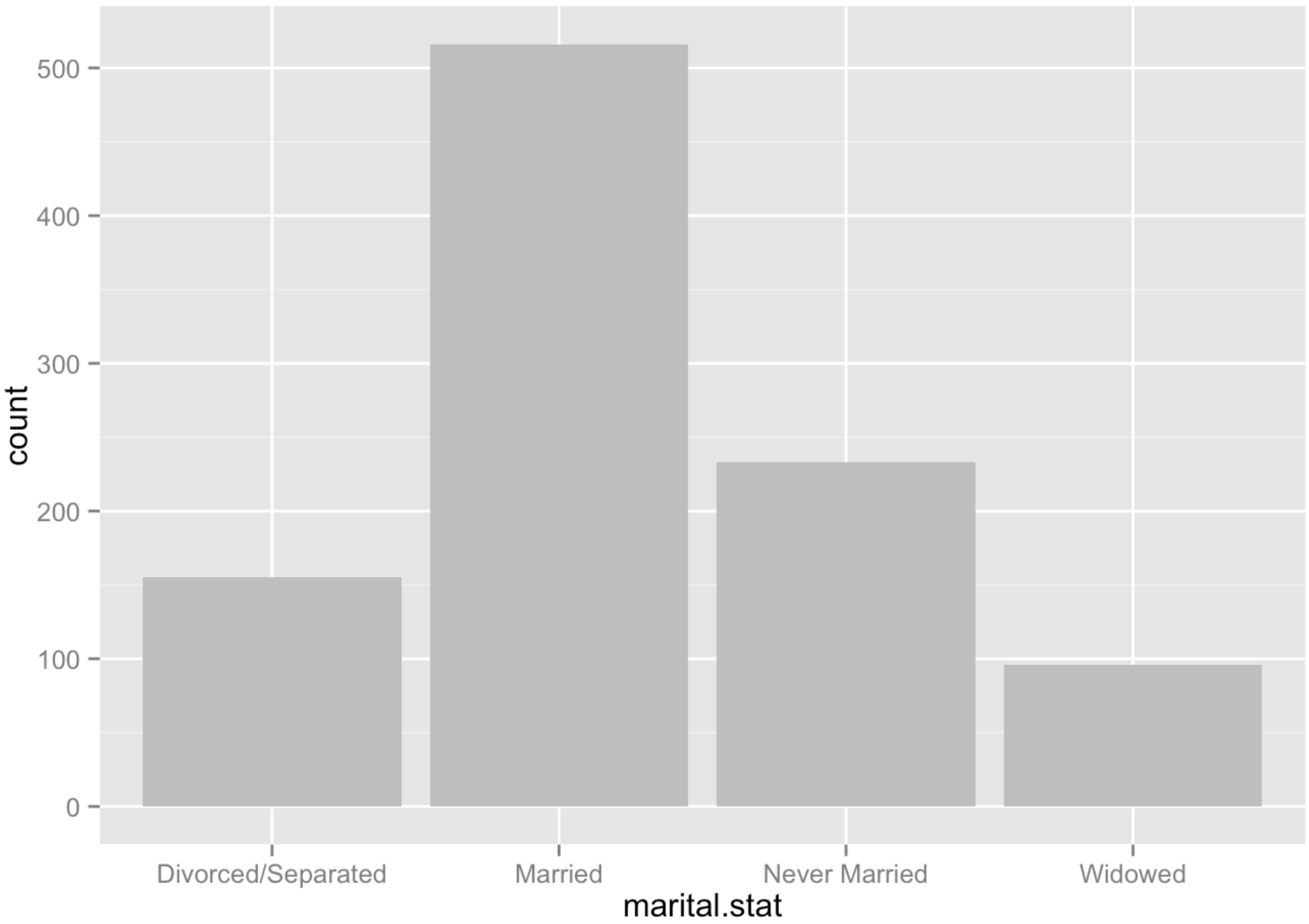
- Density plots on log-scale. 왜 `warning=FALSE` 를 켜 놓았는지 확인해 볼 것.

```
ggplot(custdata) + geom_density(aes(x=income)) +
  scale_x_log10(breaks=c(100, 1000, 10000, 100000), labels=dollar) +
  annotation_logticks(side="bt") +
  annotate("text", x=150, y=0.05, label="극히 소득이 적은 이상점", family="HCR Dotum
LVT", colour="red") +
  annotate("text", x=3000, y=0.4, label="예상을 넘는\n 1만불 대의\n 소득자들", famil
y="HCR Dotum LVT", colour="red") +
  annotate("text", x=4000, y=0.7, label="대부분의 고객은\n 2만불-10만불 수준", famil
y="HCR Dotum LVT", colour="red") +
  annotate("text", x=8000, y=0.9, label="소득분포의 정점은\n 4만불 대에", family="HCR
Dotum LVT", colour="red") +
  annotate("text", x=400000, y=0.4, label="20만불\n 이상은\n 드물지만\n이상점으로\n 보이
지는\n 않음", family="HCR Dotum LVT", colour="red")
```



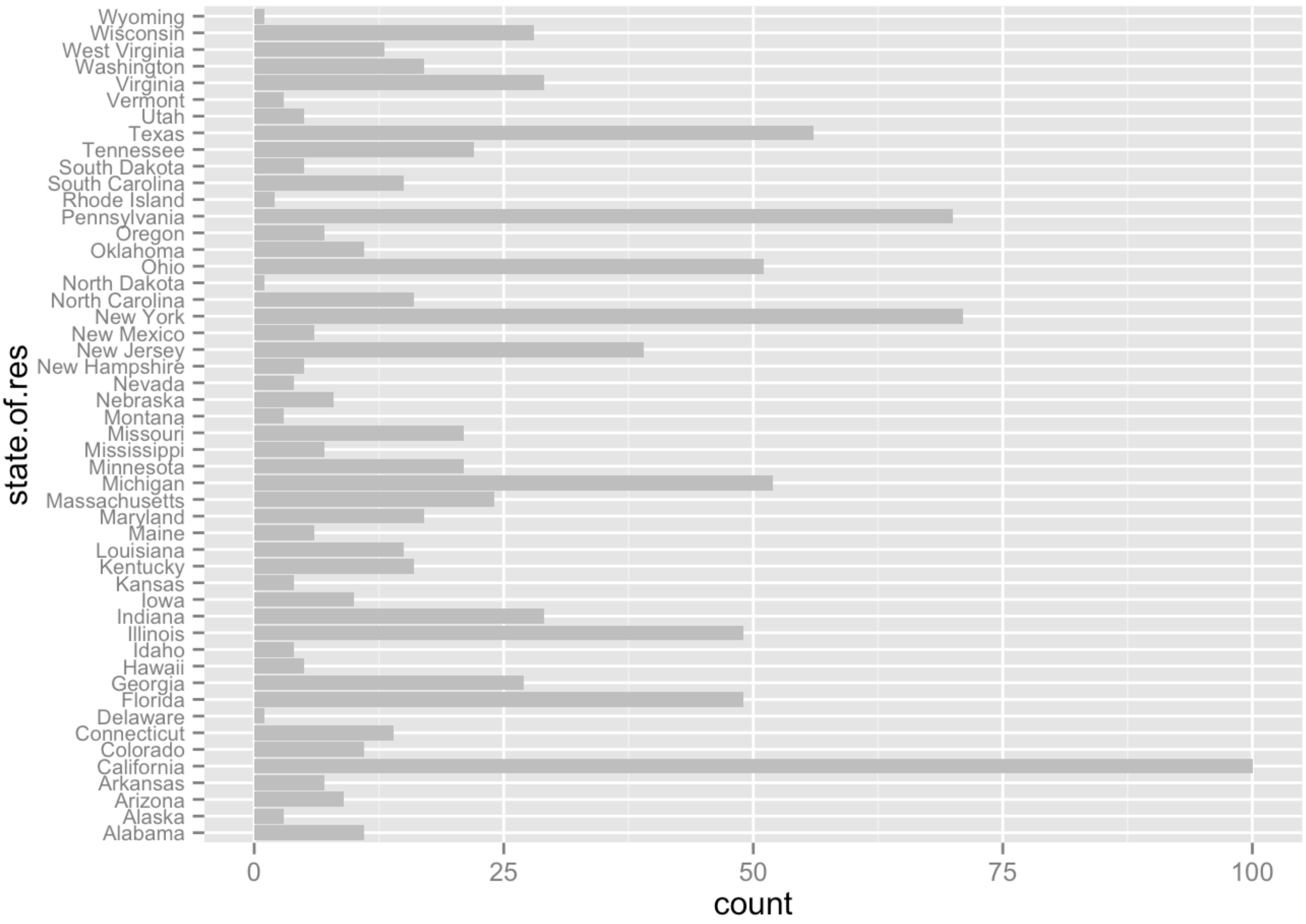
- Bar Charts

```
ggplot(custdata, aes(x=marital.stat)) + geom_bar(fill="gray")
```



- Bar Charts for `state.of.res`

```
ggplot(custdata, aes(x=state.of.res)) + geom_bar(fill="gray") +
  coord_flip() +
  theme(axis.text.y=element_text(size=rel(0.8)))
```



- 등록된 거주자 수효대로 각 주를 정렬시키려면, `reorder()` 가 필요함. 교재의 방법을 따르면 다음과 같이 할 수 있음.

```
(sor.tbl <- table(custdata$state.of.res))
```

```
## 
##        Alabama         Alaska        Arizona       Arkansas     California 
##             11              3              9              7            100 
##       Colorado    Connecticut       Delaware        Florida        Georgia 
##             11             14              1             49             27 
##         Hawaii          Idaho       Illinois        Indiana           Iowa 
##              5              4             49             29             10 
##         Kansas       Kentucky      Louisiana          Maine       Maryland 
##              4             16             15              6             17 
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri 
##             24             52             21              7             21 
##        Montana       Nebraska         Nevada  New Hampshire     New Jersey 
##              3              8              4              5             39 
##     New Mexico       New York North Carolina   North Dakota           Ohio 
##              6             71             16              1             51 
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina 
##             11              7             70              2             15 
##   South Dakota      Tennessee          Texas           Utah        Vermont 
##              5             22             56              5              3 
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming 
##             29             17             13             28              1
```

```
(sor.df <- data.frame(sor.tbl))
```

```
##                Var1 Freq
## 1         Alabama   11
## 2          Alaska    3
## 3         Arizona    9
## 4        Arkansas    7
## 5      California  100
## 6        Colorado   11
## 7     Connecticut   14
## 8        Delaware    1
## 9         Florida   49
## 10        Georgia   27
## 11         Hawaii    5
## 12          Idaho    4
## 13       Illinois   49
## 14        Indiana   29
## 15           Iowa   10
## 16         Kansas    4
## 17       Kentucky   16
## 18      Louisiana   15
## 19          Maine    6
## 20       Maryland   17
## 21  Massachusetts   24
## 22       Michigan   52
## 23      Minnesota   21
## 24    Mississippi    7
## 25       Missouri   21
## 26        Montana    3
## 27       Nebraska    8
## 28         Nevada    4
## 29  New Hampshire    5
## 30     New Jersey   39
## 31     New Mexico    6
## 32       New York   71
## 33 North Carolina   16
## 34   North Dakota    1
## 35           Ohio   51
## 36       Oklahoma   11
## 37         Oregon    7
## 38   Pennsylvania   70
## 39   Rhode Island    2
## 40 South Carolina   15
## 41   South Dakota    5
## 42      Tennessee   22
## 43          Texas   56
## 44           Utah    5
## 45        Vermont    3
## 46       Virginia   29
## 47     Washington   17
## 48  West Virginia   13
## 49      Wisconsin   28
## 50        Wyoming    1
```

```
colnames(sor.df) <- c("state.of.res", "count")
str(sor.df)
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ state.of.res: Factor w/ 50 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7 8
9 10 ...
##  $ count       : int  11 3 9 7 100 11 14 1 49 27 ...
```

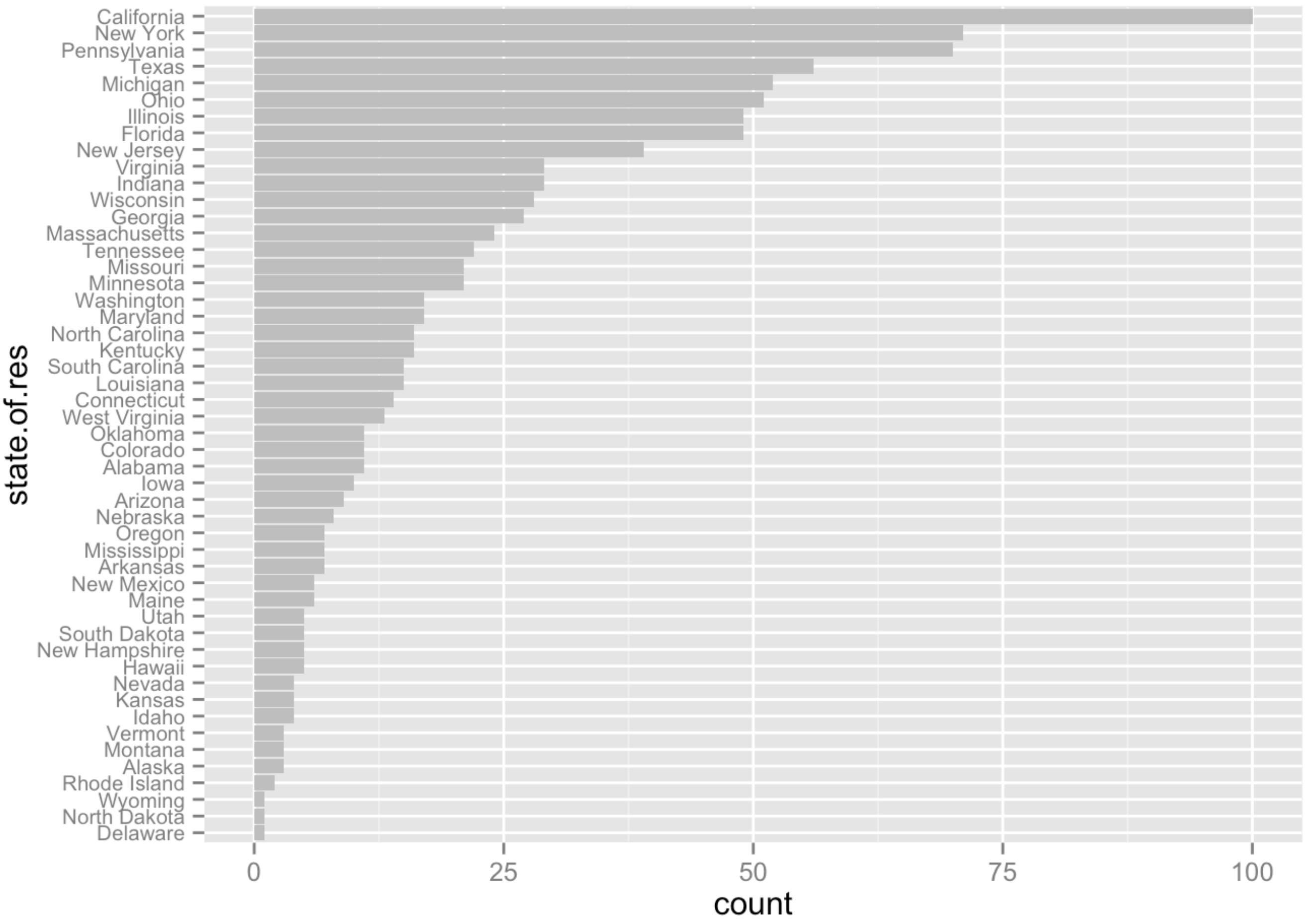
```
str(reorder(sor.df$state.of.res, sor.df$count))
```

```
##  Factor w/ 50 levels "Delaware","North Dakota",..: 23 5 21 17 50 24 27 1 43
38 ...
##  - attr(*, "scores")= num [1:50(1d)] 11 3 9 7 100 11 14 1 49 27 ...
##   ..- attr(*, "dimnames")=List of 1
##   .. ..$ : chr [1:50] "Alabama" "Alaska" "Arizona" "Arkansas" ...
```

```
sor.df.o <- transform(sor.df, state.of.res=reorder(state.of.res, count))
str(sor.df.o)
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ state.of.res: Factor w/ 50 levels "Delaware","North Dakota",..: 23 5 21 1
7 50 24 27 1 43 38 ...
##   ..- attr(*, "scores")= num [1:50(1d)] 11 3 9 7 100 11 14 1 49 27 ...
##   .. ..- attr(*, "dimnames")=List of 1
##   .. .. ..$ : chr  "Alabama" "Alaska" "Arizona" "Arkansas" ...
##  $ count       : int  11 3 9 7 100 11 14 1 49 27 ...
```

```
ggplot(sor.df.o, aes(x=state.of.res, y=count)) + geom_bar(stat="identity", fil
l="gray") +
  coord_flip() +
  theme(axis.text.y=element_text(size=rel(0.8)))
```

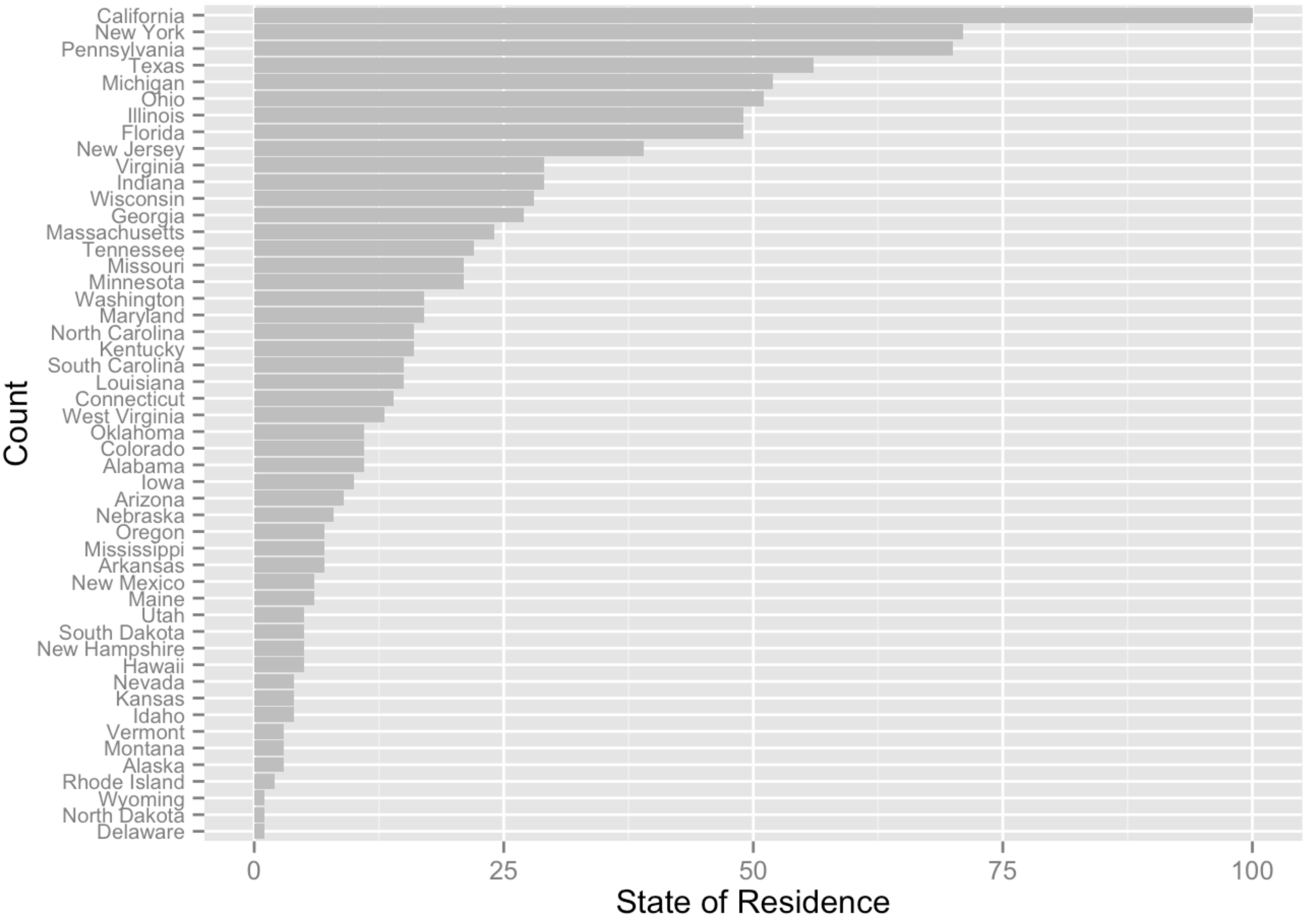


- 굳이 `transform()` 까지 사용하지 않더라도, `sor.df` 만 가지고도 원하는 작업은 할 수 있음.

```
(sor.df.2 <- data.frame(sor.tbl))
```

```
##              Var1 Freq
## 1         Alabama   11
## 2          Alaska    3
## 3         Arizona    9
## 4        Arkansas    7
## 5      California  100
## 6        Colorado   11
## 7     Connecticut   14
## 8        Delaware    1
## 9         Florida   49
## 10        Georgia   27
## 11         Hawaii    5
## 12          Idaho    4
## 13       Illinois   49
## 14        Indiana   29
## 15           Iowa   10
## 16         Kansas    4
## 17       Kentucky   16
## 18      Louisiana   15
## 19          Maine    6
## 20       Maryland   17
## 21  Massachusetts   24
## 22       Michigan   52
## 23      Minnesota   21
## 24    Mississippi    7
## 25       Missouri   21
## 26        Montana    3
## 27       Nebraska    8
## 28         Nevada    4
## 29  New Hampshire    5
## 30     New Jersey   39
## 31     New Mexico    6
## 32       New York   71
## 33 North Carolina   16
## 34   North Dakota    1
## 35           Ohio   51
## 36       Oklahoma   11
## 37         Oregon    7
## 38   Pennsylvania   70
## 39   Rhode Island    2
## 40 South Carolina   15
## 41   South Dakota    5
## 42      Tennessee   22
## 43          Texas   56
## 44           Utah    5
## 45        Vermont    3
## 46       Virginia   29
## 47     Washington   17
## 48  West Virginia   13
## 49      Wisconsin   28
## 50        Wyoming    1
```

```
ggplot(sor.df.2, aes(x=reorder(Var1, Freq), y=Freq)) + geom_bar(stat="identit
y", fill="gray") +
  coord_flip() +
  theme(axis.text.y=element_text(size=rel(0.8))) +
  xlab("Count") + ylab("State of Residence")
```


California
New York
Pennsylvania
Texas
Michigan
Ohio
Illinois
Florida
New Jersey
Virginia
Indiana
Wisconsin
Georgia
Massachusetts
Tennessee
Missouri
Minnesota
Washington
Maryland
North Carolina
Kentucky
South Carolina
Louisiana
Connecticut
West Virginia
Oklahoma
Colorado
Alabama
Iowa
Arizona
Nebraska
Oregon
Mississippi
Arkansas
New Mexico
Maine
Utah
South Dakota
New Hampshire
Hawaii
Nevada
Kansas
Idaho
Vermont
Montana
Alaska
Rhode Island
Wyoming
North Dakota
Delaware
Count
State of Residence